REMARKS

I. CLAIM REJECTIONS UNDER §112

Claims 1-9 were rejected under §112, second paragraph as being indefinite. With this Amendment, claims 1-9 are amended to clarify the terminology. None of the clarifying amendments are made in view of any prior art references.

A. **Claim 1**

With respect to the Examiner's comments regarding claim 1, the preamble has been reordered to more clearly recite a "parallel turbo decoder."

Claim 1 is also amended to clarify the connections between the interleaver memory and the routing multiplexer.

Claim 1 is further amended to clarify the connections between module inputs and outputs to the inputs and outputs of the multiplexer.

In addition, the Examiner stated, "Secondly, the limitation "... coupled to p of the n interleaver memory inputs ..." is not clear because p is defined to be the number of outputs and n is the number of inputs, where n is greater than/equal to p." This phrase has been re-worded, but was believed to be accurate. The variable "p" is simply a number. There are p multiplexer outputs. Not all of the n multiplexer inputs need to be used ($p \leq n$). Since there are p/2 modules in the first row, each with first and second input, there are p inputs that are connected to "p of the n multiplexer inputs."

Accordingly, independent claim 1 is now believed to be sufficiently clear and definite under 35 U.S.C. §112, second paragraph.

The preambles of claims 2-4 are amended to follow the amendments made to claim 1.

B. **Claim 5**

Independent claim 5 is similarly amended to clarify the module connections.

With respect to the phrase "... wherein p is an integer variable ...," the Examiner asserts that this phrase does not define p. Again, p is simply a number. If claim 5 recites "up to p memories" and "p/2 of the modules" then for any given value of p, there is a defined number of modules. So claim 5 appears to be sufficiently definite with respect to the use of the variable "p".

The preambles of claims 6-9 are amended to follow the amendments made to claim 5.

Applicants respectfully request that the rejection under §112, second paragraph, be withdrawn.

C. **Claim 21**

Claim 21 generally corresponds to claim 1 re-written to include the elements of claim 2. In addition, claim 21 is amended to better incorporate the language of claim 2 and to eliminate the interleaver memory.

II. <u>CLAIM REJECTIONS UNDER §103</u>

Claims 1 and 5 were rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over Gatherer et al. U.S. Patent No. 6,603,412 further in view of the cited Savage reference.

A. **It is Not Obvious How Gatherer et al. Could be Modified According to Savage.**

Looking at Gatherer et al., it is not obvious to a person of ordinary skill in the art how the iterative turbo coder of Gatherer et al. could be modified to incorporate the network router of Savage, and the Office Action has failed to show how such a modification could be made.

Gatherer et al. disclose a "quasi-parallel read/write intervleaver architecture" for iterative turbo coders using address contention for initiating the next data subblock, whereas Savage discloses a network router.

Even if the references were combined, certain claim elements, such as the claimed <u>memory containing a plurality of control bit tables</u>, are completely missing from both references. The Office Action fails to address this deficiency in the proposed combination.

The Office Action provides no explanation (and neither do the references) as to how the turbo coder of Gatherer et al. could be modified to include the claimed modules, how the modules would be coupled the outputs of an interleaver memory, or how the modules would be controlled in the context of a turbo decoder, as recited in claim 1.

1. Gatherer et al.

Contrary to the Office Action, Gatherer et al. do not disclose a "parallel turbo decoder" as recited in claim 1. Gatherer et al. expressly state that their coder has a "quasi-parallel read/write intervleaver architecture". (See, e.g., Abstract, col. 2, line 11, col. 3, line 25and col. 11, lines 59-60).

In contrast, the structures recited in claims 1 and 5 provide a true parallel turbo decoder, and this structure is not disclosed nor even contemplated by Gatherer et al.

Further, Gatherer et al. provide no teaching or suggestion of using a plurality of modules or a memory containing a plurality of control bit tables as recited in claims 1 and 5. Moreover, Gatherer et al. do not provide any mechanism for changing permutations and do not suggest that such a feature could or should be added in an iterative turbo decoder.

The structures recited in claims 1 and 5 allow changes in data permutations and access to memories dynamically and on the fly using control signals applied to the modules' control inputs.

2. Savage

Likewise, the Savage reference provides no teaching or suggestion of using a routing multiplexer system for interleaving data in a turbo decoder or how a turbo decoder could be so modified.

Savage does not disclose,

> "a memory containing a plurality of control bit tables each containing a plurality of control bits in an arrangement based on a respective permutation, the memory being responsive to the selected permutation to supply the plurality of control bits of the control bit table that corresponds to the selected permutation to respective control inputs of the modules.

as recited in claim 1, for example. These elements are completely lacking from both references and the proposed combination.

Further, the Savage reference does not constitute analogous art. As stated in Section 7.8 on page 309, the Savage reference is directed to the design of network routers having the task of transmitting messages among processors via nodes of a network. Network routing has nothing to do with interleavers for permuting data in a turbo decoder. A person of ordinary skill in the art would therefore not look to a network router when designing a parallel turbo decoder.

In any case, the Savage reference provides no teaching or suggestion that its permutation network could be adapted for use in a turbo code interleaver such as that shown by Gatherer et al. or how it could be done.

Although Savage states at the top of page 311 that a network routing permutation can be computed more quickly with the Benes offline permutation network, Savage also state near the bottom of page 311 that such a global permutation network in a network routing application requires switch settings that must be computed and distributed globally, which imposes important limitations on the time to realize particular permutations. Further, these characteristics appear to be directed to the application of network routing. Thus, these statements also provide no motivation for using a routing multiplexer system comprising a plurality of modules and a plurality of control bit tables for interleaving data in a turbo code interleaver.

Therefore, it would not have been obvious for a person of ordinary skill in the art to combine the teachings of Savage and Gatherer et al. Also, even if the references where combined as suggested in the Office Action, the resulting combination would still lack positively recited elements of independent claims 1 and 5.

Accordingly, Applicants respectfully request that the rejection of claims 1 and 5 under §103(a) based on Gatherer et al. and the Savage reference be withdrawn.

The Director is authorized to charge any fee deficiency required by this paper or credit any overpayment to Deposit Account No. 12-2252.

Respectfully submitted,

WESTMAN, CHAMPLIN & KELLY, P.A.

By: /David D. Brush/
David D. Brush, Reg. No. 34,557
900 Second Avenue South, Suite 1400
Minneapolis, Minnesota 55402-3319
Phone: (612) 334-3222 Fax: (612) 334-3312

DDB/tkj